Homework 4 – Automated Deployment with Ansible

ECE 530 – Introduction to Cloud Computing

Instructors: I. Papapanagiotou

Introduction

The purpose of this Homework exercise is to introduce you to the use of automation and DevOps tools. So far, you have learned how to provision virtual machines to host applications in the cloud. Automation tools allow you to develop a standard configuration that can be applied to machines when they are created, making it easy to stamp out new instances of a cluster easily or to make a VM operational quickly after it has been destroyed. While there are many automation tools available for different use cases, this exercise will focus on the development of an **Ansible Playbook**.

What is Ansible

Ansible is an open-source software provisioning, configuration management, and application-deployment tool that runs on Linux. Ansible includes its own declarative language and uses playbooks that create the environment for running simple or incredibly complex commands (or groups of commands) on a remote machine.

Goal

In the previous homework, we learned on how to spin a MongoDB cluster in Docker containers and semi-automate it with a Dockerfile. In this homework, we are going to build on top of that and use an ansible playbook to configure 3 VMs (on your laptop or AWS) or Docker containers (which might be the easiest) as MongoDB cluster (one primary and two secondaries). gives the v Your playbook should be written in a way such that it uses group_vars and templates to avoid using hard-coded variables in configuration files.

Sample Install Ansible Package

If you don't already have Ansible running on one of your servers, follow these steps to install it:

1. Log into the Ubuntu Server that will host Ansible
2. Install the necessary repository with the command *sudo apt-add-repository ppa:ansible/ansible*.
3. Update apt with the command *sudo apt-get update*.
4. Install Ansible with the command *sudo apt-get install ansible -y*.
5. If necessary, install a Python interpreter with the command *sudo apt-get install python -y*.

Deliverable

A zip file of an Ansible Playbook. The playbook must contain a README with your team member names and instructions on how to run the playbook. The playbook must:

- Must be written for Ansible 2.5 or newer

- Have an inventory with two groups, the primary and secondary groups
- Have a common role that applies to all nodes.
- Have separate roles for the master and secondary nodes.
- Use variables in group_vars and templates to create generic playbooks. You must have variables for at least 2 different secondary nodes.
- Be successful in a single run against a fresh VM

Instructions

Read the Ansible guide around creating a playbook:
https://docs.ansible.com/ansible/intro_getting_started.html

Use the grading rubric below for suggestions on the order of tasks to complete to be most successful.

Grading Rubric

You will be graded on your playbook's ability to complete the following tasks.

Task	Points
Create a playbook with a common role that writes your team member's names to the file /root/team.txt on all nodes	30
Create a role that provisions a MongoDB primary node • Creates config files • Starts the MongoDB service	20
Make your Primary Node role generic to use variables and templates instead of hard-coded config files	10
Create a role that provisions at least one MongoDB secondary node • Creates config files • Starts the MongoDB service	20
Create a generic MongoDB secondary node role that uses variables in group_vars and templates to customize each node. Must provide variables for at least two secondary nodes	20
EXTRA CREDIT: When configuration on the primary or secondary changes, a handler is used to restart the daemon.	10

If your playbook runs with errors, 10 points will be deducted from your score.